EPFL
VITA

Report
Bachelor Semester Project

Taking the Driving Theory Test with Vision-Language Models

by Matthias Wyss (SCIPER 329884)
Communication Systems, BA6
8 ECTS

Supervised by
Prof. Alexandre Alahi
Dr. Charles Corbière

Contents

1 Introduction

The DrivingTheory project aims to evaluate the capacity of vision-language models to pass driving theory exams. These exams are usually composed of 40 multiple-choice questions, each presenting unique challenges that test knowledge of driving theory, rules, and situational judgment. We provide the model with images, questions, possible answers, and explanations for training, and expect it to output the correct answers, which makes the task closely related to visual question answering. The challenge lies in fine-tuning a model that not only understands textual questions but also interprets the relevant elements in the associated images to select the correct answers. This fusion of vision and language understanding is pivotal in real-world scenarios where drivers must quickly assess and respond to visual stimuli on the road. By evaluating the efficacy of vision-language models in this context, we aim to bridge the gap between artificial intelligence capabilities and practical driving skills. This project is a collaboration between VITA lab and NLP lab. This semester, we were 2 students working fully on this project. My job was to get more data that would be useful to train a model that solves driving theory tests. Max Conti worked on the fine-tuning and evaluating part.

2 Related Work

2.1 State of the Driving Theory Project

The project was started last semester by Pierre Ancey. Therefore, I started my work by picking up where Pierre left off [1]. He scraped 3 French websites: `CodeRoute`, `PasseTonCode` and `Ornikar` to form a dataset of 3'367 unique problems with images. Then he evaluated them [2] with LaVIN [3], a vision-language model, and got around 69% of correct answers, which is not sufficient to pass the test.

2.2 BERT

BERT [4] (Bidirectional Encoder Representations from Transformers) is a pre-trained language model developed by Google. Known for its bidirectional training, BERT understands the context of words from both preceding and succeeding words, unlike traditional unidirectional models. This capability is vital for tasks like question answering, sentiment analysis, and text classification.
In our project on theoretical driving exams, BERT classified dataset samples into one of the 10 official themes without explicit labels. Its effectiveness in understanding and categorizing these themes demonstrates its ability to handle complex textual data typical of driving theory exam problems.

3 Method Implemented

3.1 Dataset

As said above, there was an existing dataset of 3'367 samples with images. I managed to triple this dataset to reach 10'046 unique samples with images. I also produced another dataset of 145 unique samples with videos, which could be useful for future implementations. These 2 datasets are stored on the RCP. The code is available on the `vita-epfl` GitHub on this repo [5].

3.1.1 Scraping

In order to enlarge the dataset, I scraped three additional French websites: `EnVoitureSimone` [6], `CodeClic` [7], and `Stych` [8]. In the following sections, I will detail the specifications of each website and explain my methodology.

EnVoitureSimone
For `EnVoitureSimone`, I initially focused on scraping "Examens blancs," which consist of 45 series of 40 questions each. This method ensured I scraped all the questions, but the themes of the questions were not captured. Recognizing the importance of themes for future dataset splits, I shifted my approach to scrape thematic series. There are 11 themes (including "Autres"), and each series consists of 40

randomly selected questions from a specific theme, meaning not all questions could be guaranteed to be scraped.
To scrape `EnVoitureSimone`, I had to deal with each question individually. After running the scraper for over 24 hours, I collected 1'472 samples samples with images and 145 samples with videos. I separated the samples with videos in an other file, since for now we were using only images.

CodeClic
For `CodeClic`, there are two ways to obtain questions: through "Tirage aléatoire", which provides a random series of 40 questions, and "Test thématique", which offers 40 questions randomly selected from one of 11 themes. I chose to scrape using the thematic tests to capture the themes along with the questions. This method allowed me to scrape all data from a series in one go, making it faster.
After running the scraper for a few hours, I collected 2'750 questions. Each sample contains either an image or a `GIF` (convertible to `JPEG` without loss of information).

Stych
For `Stych`, the questions come from different sources:

- Cours (11 themes)
  - Cours vidéo: Watch a video and answer 5 questions on the theme.
  - Cours fiche: Read a text and answer 5 questions on the theme.
- Test: Series of 20 random questions from a specific theme.
- Examens blancs: 25 series of 40 questions each, with unknown themes.

I scraped all the questions from "Cours vidéo" and "Cours fiche", capturing the themes. Then, I scraped the "Examens blancs", although I could not capture the themes. I also let the code run to scrape the thematic tests for a day. Ultimately, I collected 2'457 samples with images.

3.1.2 Processing Scraped Data

I merged the datasets from Pierre and the three newly scraped datasets into a big `JSON` file of 10'046 samples with images. However, each site had different themes with varied names. To harmonize the themes according to the 10 official themes, I followed the table in appendix A. Some issues arose: `Stych` had no samples fitting into the "Les notions diverses" category, and some samples from `Stych`, `CodeRoute`, and `PasseTonCode` lacked themes. To ensure consistent data splits, respecting the number of questions per theme for each series of 40 questions, I implemented a theme classifier.

3.1.3 Theme Classifier

The challenge was to classify questions without themes into one of the 10 official themes. As input for training I used for each samples the question(s), possible answers and explanation. The output is the predicted theme among the 10. Using approximately 6'000 questions with themes for training and 4'000 questions to classify, I had the opportunity to fine-tuned on a GPU of the RCP, several BERT based models: `bert-base-cased` achieved 76.5% accuracy, `flaubert_base_cased` 77.6%, and `camembert-base` 80.6%. Despite attempts with other models and ensemble methods, none outperformed `camembert-base`.
Other methods such as KNN (77.2%), SVM (78.8%), Random Forest (73.7%), CNN (77.2%), K-means (33.5%), Naive Bayes (54.9%) and GMM (11.9%) were also tested, but `camembert-base` remained the best performer. Although 80% accuracy was not sufficient, a confusion matrix analysis revealed difficulty in distinguishing similar themes. I also tried to weight probabilities with the global theme distribution for `camembert-base` but it improved accuracy by only 0.5%. By uniquely classifying samples with $\geq 90\%$ certainty, I achieved 50% classification coverage, though some themes had no classified samples. Using SVM with TF-IDF, incorporating stop words, allowing n-grams, lemmatizing, and stemming the inputs resulted in 82.5% accuracy, making this the best choice. We can see

on appendix B that the classifier struggles to classify samples between similar themes. The code is available on the `vita-epfl` GitHub on this repo [5]. Knowing the theme of each sample in the dataset allows us to split the dataset while maintaining the official distribution of themes. Additionally, it enables us to produce more precise metrics by theme and potentially leverage this information in training that takes context into account.

3.1.4 Series Building and Splits

The french driving theory test is composed of 40 multiple-choice questions. We decided to split the data into 3 splits:

- A validation set of 880 questions (9% of the dataset): 6 series from `CodeClic`, 5 from `Stych`, 4 from `Ornikar`, 3 from `EnVoitureSimone`, 2 from `CodeRoute`, and 2 from `PasseTonCode`
- A test set of 880 questions (9% of the dataset): 6 series from `CodeClic`, 5 from `Stych`, 4 from `Ornikar`, 3 from `EnVoitureSimone`, 2 from `CodeRoute`, and 2 from `PasseTonCode`
- A train set of 8'286 questions (82% of the dataset). No need for series here.

The goal was to build series of 40 questions while maintaining the official theme distribution from the exam described in appendix C. For `PasseTonCode` and `Stych`, I applied the theme classifier and for these 2, `CodeClic`, `Ornikar` and `EnVoitureSimone`, I created series split with respect to the predicted theme and the theme distribution. Manual adjustments were made for `CodeClic` due to an insufficient number of samples in some themes. For `CodeRoute`, series were built by identifying series IDs with 40 samples, following Pierre's method, also due to an insufficient number of samples in some themes.

3.1.5 Statistics

You can find some statistics about the dataset in appendix D.

3.1.6 New results

We have now a big dataset from 6 different sources that ensure more diversity on the questions and images. All the samples have an associated theme, that is also useful to derive more precise metrics. Max Conti evaluated this new dataset with LLaVA [9] and got a score of 22/40. With the old one, he got 19.5/40. Therefore with this new dataset, we increased the score by 2.5 points.

3.2 Other Data

3.2.1 Compilation of Things to Know

I created a file named `things_to_know.txt` that compiles essential information candidates need to know by heart before taking the driving theory test. This file consolidates crucial points sourced from various websites. It is stored on the RCP. Including this information could be beneficial for the model if used as contextual data, providing it with a richer understanding of the key facts and rules that are critical for the driving theory test.

3.2.2 Generating Questions and Answers from Legal Documents

I attempted to generate questions and answers from the official French highway code legal documents. First, I compiled the legal texts into a `.txt` file. Then, I used a questions and answers generation model [10], on these texts. However, this approach proved to be less effective because the legal documents contain technical jargon, and the model, trained on English data, produced outputs in a mix of French and English. Additionally, the generated questions were often not practical, such as those asking for specific legal references, e.g., "What is the the law in which...". This mismatch in language and context resulted in generated content that was not very useful.

3.2.3 Driving Theory Book Extraction

I downloaded the book `Le-code-de-la-route-pour-les-nuls-2023-24` [11] in `PDF` format with the goal of extracting the text and images to associate them accurately. Initially, I tried using GROBID [12] for extraction, but it didn't work well as it is not designed for non-research papers. Consequently, I extracted the text and images page by page. To insert images at the correct places within the text, I attempted to use CLIP similarity [13]. However, this method yielded poor results due to the low quality of the extracted text, which included many line breaks. To resolve this, I manually positioned the images within the text, a task that took me two full days. The final output is a `book.txt` file containing the text with image names in brackets (e.g., `[image1.png]`) placed appropriately, along with a folder of images. It is stored on the RCP. This will likely be useful for providing context, deriving some new dataset or for Retrieval-Augmented Generation.

4 Conclusion and Future Work

To conclude, there is a summary of what I produced:

- A dataset of 10'046 French questions with images and themes
- A dataset of 145 French questions with videos and themes
- A `.txt` document about things to know before taking the French driving theory test
- A `.txt` document of a French driving theory book with images integrated in the text

For future work, these `.txt` documents will likely be useful for the model as context, or for creating a new dataset. It would be also great to use the dataset with video-questions, for example by extracting frames from the videos. An other idea would be to use the theme classifier to provide more precise informations to the model based on the theme of the questions.
In this project, I learned a lot of things such as scraping, BERT based model fine-tuning, docker, how to use a GPU on the RCP, data processing, pdf parsing, etc.
I want to thank my supervisor, Charles Corbière, as well as Syrielle Montariol, Simon Roburin, Pierre Ancey, Max Conti, and Francesco Pettenon, who form the team of the DrivingTheory project.

Appendix

A Mapping Themes Table

Theme	Ornikar	EnVoitureSimone	CodeClic	Stych
La circulation routière	La circulation routière.	Législatif	La circulation & Les panneaux	Règles de circulation & La signalisation
Le conducteur	Le conducteur.	Conducteur	Le conducteur	Conducteur / conductrice
La route	La route.	Réglementation	La route	La route & Les intersections
Les autres usagers	Les autres usagers.	Usagers	Les autres usagers	Autres usagers
Les notions diverses	Les notions diverses.	Divers	Les notions diverses	?
Les premiers secours	Les premiers secours.	Aides	Les secours	Réglementation et accidents
Prendre et quitter son véhicule	Prendre et quitter son véhicule.	Poste de conduite	Prendre et quitter son véhicule	Entrée et sortie
La mécanique et les équipements	La mécanique et les équipements.	Mécanique	La mécanique et les équipements	Éléments mécaniques
La sécurité du passager et du véhicule	La sécurité du passager et du véhicule.	Sécurité	La sécurité	Équipements de sécurité
L'environnement	L'environnement.	Ecologie	L'environnement	Éco-conduite

Table 1: Mapping Themes Table

B SVM Confusion Matrix

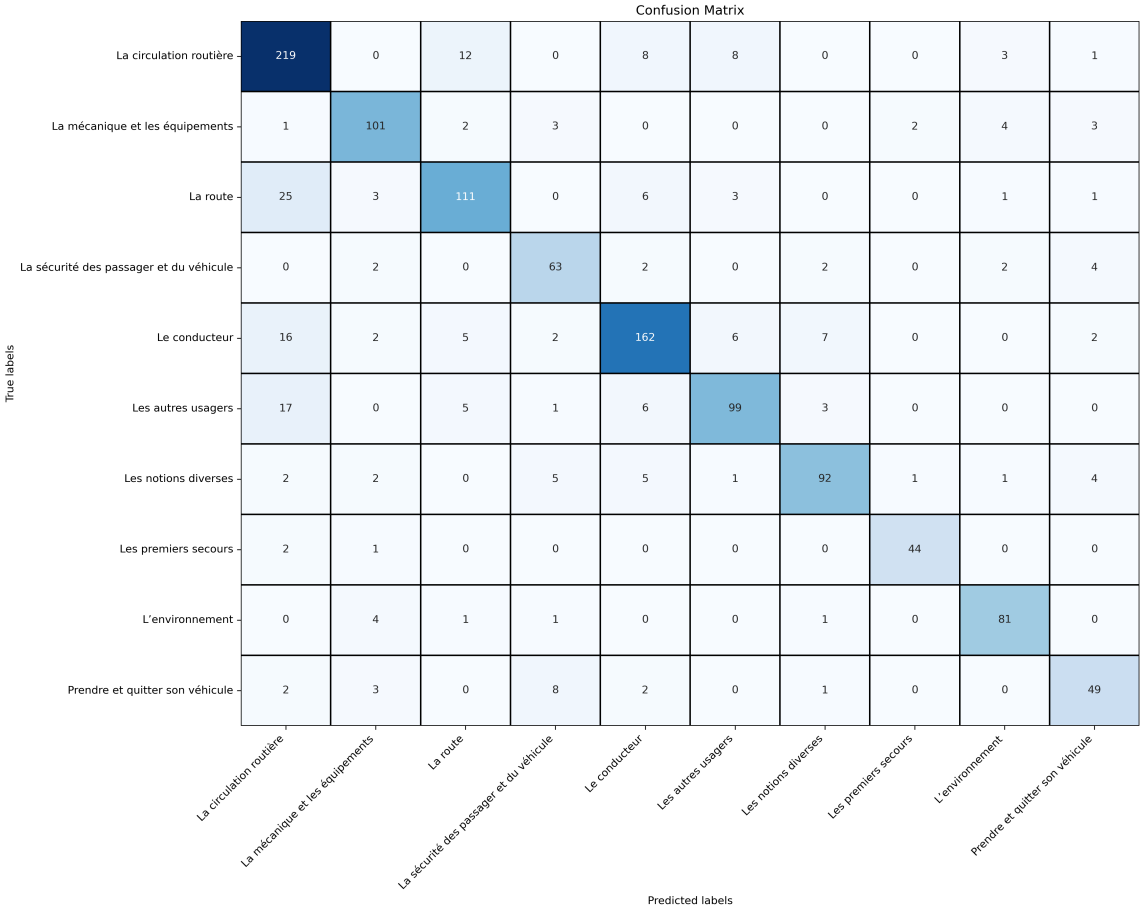

Figure 1: SVM Confusion Matrix

C Theme Distribution in a Serie

Theme	# of questions per series
La circulation routière	5
Le conducteur	8
La route	6
Les autres usagers	5
Les notions diverses	4
Les premiers secours	1
Prendre et quitter son véhicule	2
La mécanique et les équipements	3
La sécurité des passager et du véhicule	3
L'environnement	3

Table 2: Theme Distribution in a Serie

D Dataset Statistics

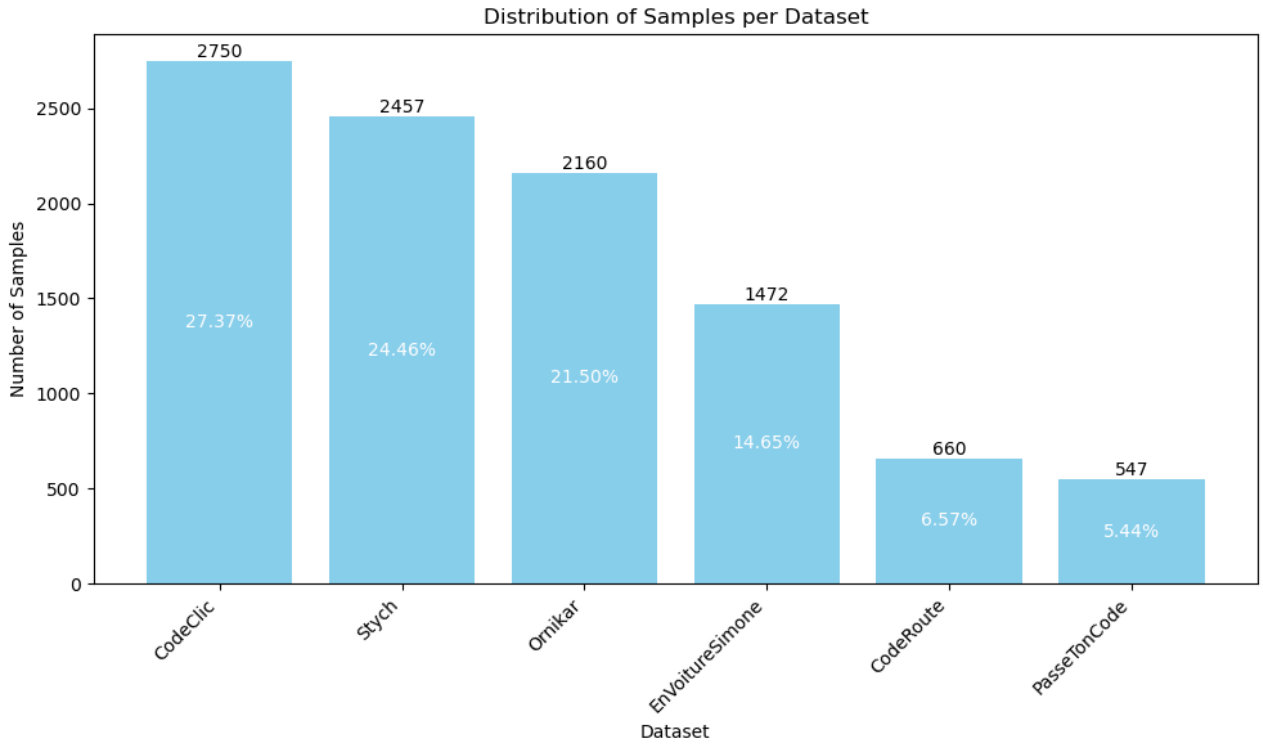

Figure 2: Distribution of Samples per Dataset

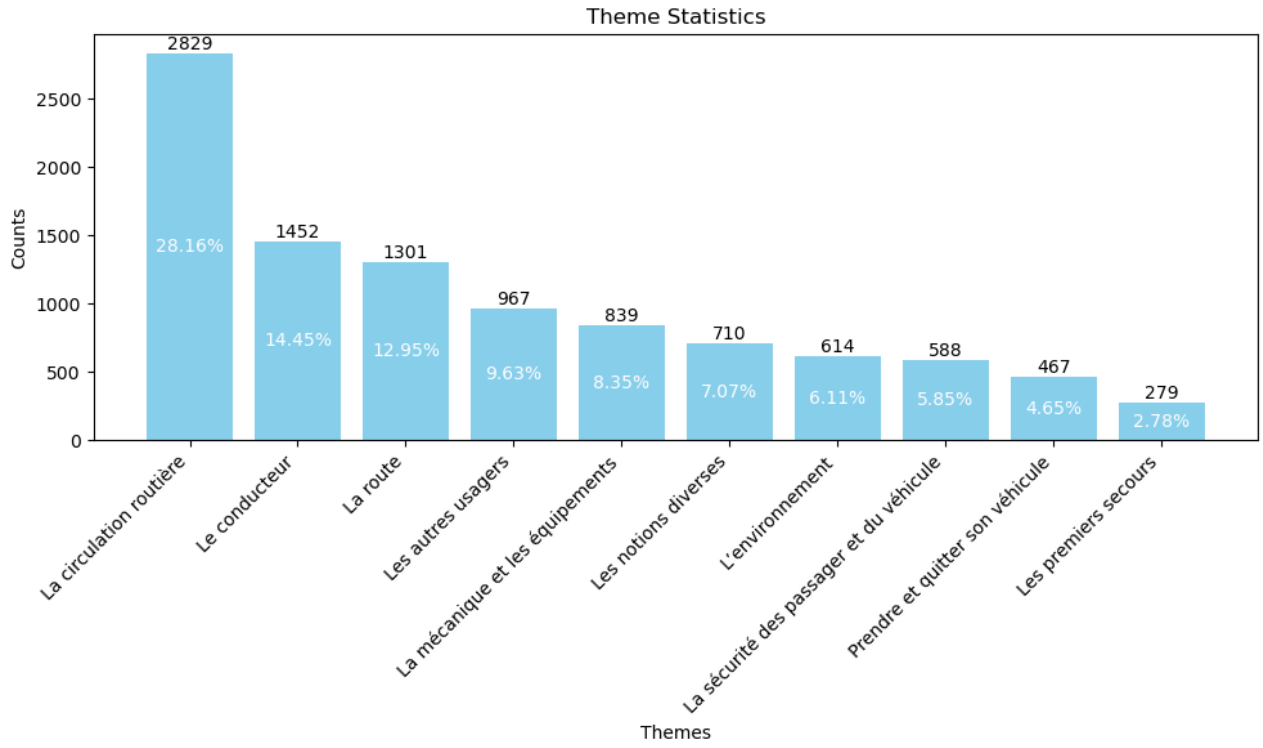

Figure 3: Themes Statistics

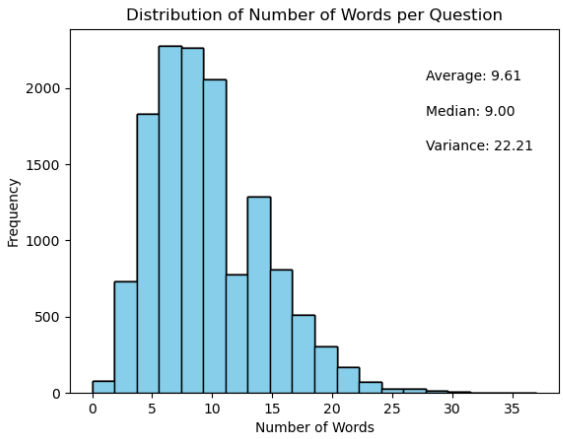

Figure 4: Distribution of Number of Words per Question

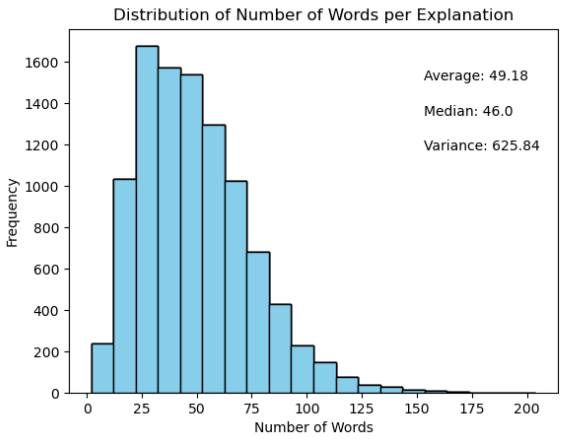

Figure 5: Distribution of Number of Words per Explication

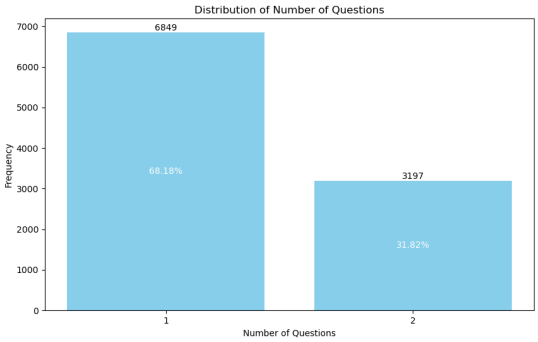

Figure 6: Distribution of Number of Questions

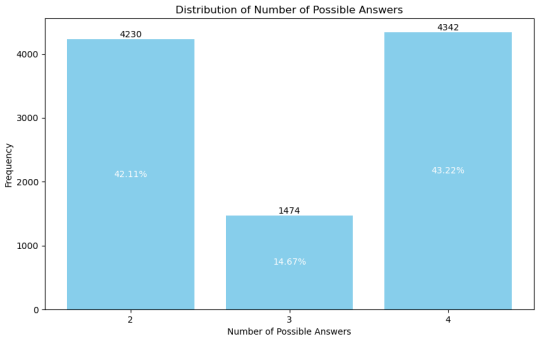

Figure 7: Distribution of Number of Possible Answers

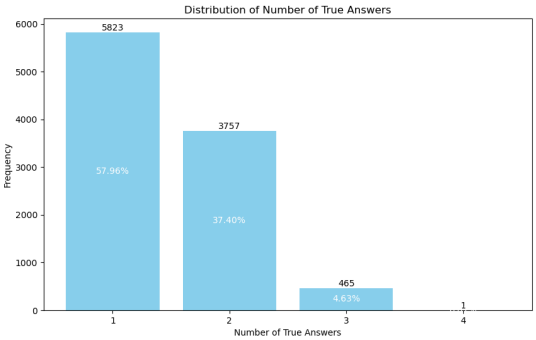

Figure 8: Distribution of Number of Possible True Answers

References

[1] Pierre Ancey. *ScrappingSemesterProject*. https://github.com/vita-epfl/ScrappingSemesterProject. 2024.

[2] Pierre Ancey. *DrivingTheory-LaVIN*. https://github.com/vita-epfl/DrivingTheory-LaVIN. 2024.

[3] Gen Luo et al. *Cheap and Quick: Efficient Vision-Language Instruction Tuning for Large Language Models*. 2023. arXiv: 2305.15023 [cs.CV].

[4] Jacob Devlin et al. *BERT: Pre-training of Deep Bidirectional Transformers for Language Understanding*. 2019. arXiv: 1810.04805 [cs.CL].

[5] Matthias Wyss. *data_extraction_driving_theory*. https://github.com/vita-epfl/data_extraction_driving_theory. 2024.

[6] *EnVoitureSimone*. Accessed: 05-06-2024. URL: https://www.envoituresimone.com.

[7] *CodeClic*. Accessed: 05-06-2024. URL: https://www.codeclic.com.

[8] *Stych*. Accessed: 05-06-2024. URL: https://www.stych.fr/code-de-la-route.

[9] Haotian Liu et al. *Improved Baselines with Visual Instruction Tuning*. 2024. arXiv: 2310.03744 [cs.CV].

[10] Potsawee Manakul, Adian Liusie, and Mark JF Gales. "MQAG: Multiple-choice Question Answering and Generation for Assessing Information Consistency in Summarization". In: *arXiv preprint arXiv:2301.12307* (2023).

[11] *Le Code de la Route pour les Nuls 2023-24*. https://www.pourlesnuls.fr/livres/le-code-de-la-route-2023-2024-poche-pour-les-nuls-9782412086254. 2023.

[12] *GROBID*. https://github.com/kermitt2/grobid. 2008–2024.

[13] Alec Radford et al. *Learning Transferable Visual Models From Natural Language Supervision*. 2021. arXiv: 2103.00020 [cs.CV].